File permissions in Linux

Project description

Through the Linux kernel, using the command line terminal, I will examine the existing permissions of user *researcher2* in the directory */home/researcher2/projects/* to determine if their authorization level matches their current file and directory permissions. If they do not match, I will use Linux commands to change permissions as required for each file.

Check file and directory details

Using the *ls -la* command in */home/researcher2/projects/* I could see the permission levels of the directory and each regular and hidden file.

```
researcher2@b80774e76762:~$ pwd
/home/researcher2
researcher2@b80774e76762:~$ ls
projects
researcher2@b80774e76762:~$ cd projects
researcher2@b80774e76762:~/projects$ pwd
/home/researcher2/projects
researcher2@b80774e76762:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 May  2 19:45 .
drwxr-xr-x 3 researcher2 research_team 4096 May  2 20:39 ..
-rw--w---- 1 researcher2 research_team   46 May  2 19:45 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 May  2 19:45 drafts
-rw-rw-rw- 1 researcher2 research_team   46 May  2 19:45 project_k.txt
-rw-r----- 1 researcher2 research_team   46 May  2 19:45 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 May  2 19:45 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 May  2 19:45 project_t.txt
```

Describe the permissions string

- **Type of content** (1st character of permission string): Directory = “D”; File = “-”
- **Types of file permission** (characters 2-10 of the permission string is in groups of 3, in respect to the type of file owner)
    - Each group is going to look like “r” or “-”, “w” or “-”, and “x” or “-”
        - Examples:
            - rwx

        - rw-
        - r-x
        - -wx
        - r--
        - -w-
        - --x
    - *Read = "r"; Write = "w"; Execute = "x"; No permissions = "-"*
- **Types of file owners:** User (*character 2,3,4*) = "u"; Group (*character 5,6,7*) = "g"; Other (*character 8,9,10*) = "o"

Example: permissions for file= *project_k.txt was*

```
-rw-rw-rw-
```

(1)(2)(3)(4)

1. "-" is the 1st character and means this is a file rather than a directory(folder). If it was a directory it would be a "D" instead.
2. "rw-" is the first group of 3 (characters 2,3,4) for the 1st owner type (User). This means the the User can read(r) and write(w) to this file, but cannot execute(x) which is why there is a dash(-) for the last of the three characters. A dash(-) means no permission and will always replace a read(r), write(w), or execute(x) character when necessary.
3. "rw-" is the second group of 3 (characters 5,6,7) of the 2nd owner type (Group). This means the Group owner type has read(r) and write(w) permissions but cannot execute("x" turns into "-")
4. "rw-" is the last group of 3 (characters 8,9,10) for the Other owner group. This group can read(r) and write(w) the file but cannot execute(-)

Change file permissions

The change permissions of each file for each owner type, the command *chmod* was used followed by the change to add(+) or remove(-) read(r), write(w), execute(x) or no permissions(-) of the file. The organization requires the 'other' group to have no write access to any files.

[Before]

```
-rw-rw-rw- 1 researcher2 research_team   46 May  2 19:45 project_k.txt
```

[Command]

```
researcher2@b80774e76762:~/projects$ chmod o-w project_k.txt
```

[After]

```
-rw-rw-r-- 1 researcher2 research_team   46 May  2 19:45 project_k.txt
```

Change file permissions on a hidden file

Using the command *ls -la* I can see the hidden file *.project_x.txt* in which no file owners types should have write(w) permission, however, the user(u) and group(g) owner types do need read(r) permissions. Everything else is no permissions(-).

[Before]

```
-rw--w---- 1 researcher2 research_team   46 May  2 19:45 .project_x.txt
```

[Command]

```
researcher2@b80774e76762:~/projects$ chmod u-w,g-w,g+r .project_x.txt
```

[After]

```
-r--r----- 1 researcher2 research_team   46 May  2 19:45 .project_x.txt
```

Change directory permissions

The organization requires that the only user who should have access to the *~drafts* sub-directory within the *~projects* parent directory. All permissions were removed from all applicable owner types.

[Before]

```
drwx--x--- 2 researcher2 research_team 4096 May  2 19:45 drafts
```

[Command]

```
researcher2@b80774e76762:~/projects$ chmod g-x drafts/
```

[After]

```
drwx------ 2 researcher2 research_team 4096 May  2 19:45 drafts
```

Summary

The directory and file permissions have been successfully modified to match the authorization levels. Permissions were added(+) and removed(-) from each owner type (user, group, other) for a variety of regular files, one hidden file, and one directory.